

IMPERIAL COLLEGE LONDON

TIMED REMOTE ASSESSMENTS 2020-2021

MSc Computing  
for Internal Students of the Imperial College of Science, Technology and Medicine

PAPER COMP70036=COMP97074

OBJECT ORIENTED DESIGN AND PROGRAMMING

Friday 30 April 2021, 10:00

Duration: 90 minutes

Includes 0 minutes for access and submission

*Answer ALL TWO questions*

Open book assessment

This time-limited remote assessment has been designed to be open book. You may use resources which have been identified by the examiner to complete the assessment and are included in the instructions for the examination. You must not use any additional resources when completing this assessment.

The use of the work of another student, past or present, constitutes plagiarism. Giving your work to another student to use constitutes an offence. Collusion is a form of plagiarism and will be treated in a similar manner. This is an individual assessment and thus should be completed solely by you. The College will investigate all instances where an examination or assessment offence is reported or suspected, using plagiarism software, vivas and other tools, and apply appropriate penalties to students. In all examinations we will analyse exam performance against previous performance and against data from previous years and use an evidence-based approach to maintain a fair and robust examination. As with all exams, the best strategy is to read the question carefully and answer as fully as possible, taking account of the time and number of marks available.

Paper contains 2 questions

- 1 Your task is to implement a system that manages real-estate property viewings. To that end, the system shall connect clients (potential tenants), real estate agents and properties for appointments and store them in a database.
- agents have a phone number and a name
  - agent phone numbers assumed to be unique, i.e., agents can be identified by their phone number
  - clients merely have a phone number that identifies them
  - properties have a latitude and longitude that identifies them
  - appointments connect an agent, a property and a client
  - appointments have a time that is stored as a simple integer (the function `now()` returns the current time)
- a Define and implement classes for the scenario above (in C++) in the file `answer1.cpp`. Select appropriate types for the class members as well as the parameters for constructors, functions and operators. You do not have to define getters and setters but may expose data members publicly.
- b In the skeleton file, you will find the declaration of a class called `DB`. This class declares four member functions: `getProperty`, `getAgent`, `getClient` and `makeAppointment` that operate on the database (four lists that store the respective entities). To see their use, consider the calls in the `main` function. Implement these functions such that the three `get...` functions return an entity if it exists in the list and create a new one if it does not (for your convenience, the skeleton file contains documentation for two functions of `std::list` that will help you: `push_back` and `back`).
- c Declare and implement `cancel()` as a member function of `Appointment` to allow the cancellation of an appointment. The `cancel` function shall determine a new time that the property, the client and the agent are available by searching the database and returning a new appointment. To support `cancel()`, implement the utility functions `DB::clientIsAvailable`, `DB::agentIsAvailable` and `DB::propertyIsAvailable` to determine if the respective entities are already involved in appointments at a given time.

*The three parts carry, respectively, 30%, 35%, and 35% of the marks.*

2 Consider the following scenario of storing and administering vaccines:

- Vials are small bottles, which can store a specific amount of fluid (in ml). When a vial is created, it is automatically filled with a vaccine (in ml). The amount of vaccine it is filled with depends on the type of vaccine it holds. Some types of vaccines need to be diluted before they can be administered safely. Therefore an amount of liquid (in ml) can be added to a vial. When a patient is vaccinated, a dose of vaccine (in ml) is taken from a vial. The dose depends on the type of vaccine and administering any other amount than the specified dose is not safe. The whole dose has to be taken from one vial and any leftover vaccine in a vial cannot be transferred to another vial.
  - A vaccine has a name, the amount of vaccine (in ml) that is stored in a vial when it is created, and the dose. There are two types of vaccines, type A and type B. Type A vaccines must be diluted by a specified amount of liquid before they can be administered safely while type B vaccines must not be administered above a maximum temperature (called safe temperature), which is 20 degrees Celsius if not specified.
  - Vials can be placed on and removed from trays, which have a size that specifies how many vials they can hold. At any moment it shall be possible to print information about the vials and the vaccines they contain on a tray to standard output. This information shall include the name of the vaccines, their doses, by how much they have to be diluted (if applicable), what their safe temperatures are (if applicable), and how much liquid is left in the vials.
  - Use the given template class *vector* as the underlying container, for which the relevant declarations can be found in the provided skeleton file *answer2.cpp*. You can assume that a function *float getTemp()* exists that returns the current temperature. The unit of all amounts of liquids is milliliter (ml).
- a Write C++ class declarations and definitions to support the scenario above in the file *answer2.cpp*. Ensure that the *tray* is implemented as a class template and specify the *maximum number* of objects a tray can hold through a template parameter.
- b Add code to *int main()* in *answer2.cpp* that does the following:
- Create two vials. The first vial can hold a maximum of 3ml of liquid and the second vial can hold a maximum of 6ml of liquid. They come with the following vaccines:

Vial	Type	Name	Vaccine in Vial	Dose	Dilution	Max. Temp.
1	A	Covid-Protect	0.45 ml	0.3 ml	1.8 ml	N/A
2	B	Covid-Begone	5 ml	0.5 ml	N/A	15°C

N/A means not applicable.

- Incorrectly dilute the vaccine in the first vial by adding only 1.1 ml of liquid.
- Create a tray that can hold 10 vials and place both vials on it.
- Take one vial from the tray and use this vial to administer doses of the vaccine in this vial for as long as it is safe to do so. Once it is not safe anymore, discard the current vial and take the next from the tray. Continue until there is no more vial on the tray. Print information about vials and their vaccines that are on the tray every time before administering a dose of vaccine.

*The two parts carry, respectively, 80% and 20% of the marks.*